3D Maquetter: Sketch-based 3D Content Modeling for Digital Earth

Kaveh Ketabchi, Adam Runions, Faramarz F. Samavati
Department of Computer Science
University of Calgary, Calgary, Alberta, Canada
{kketabch, runionsa, samavati}@ucalgary.ca

***Abstract*—We present a sketch-based system for the creation and editing 3D content such as Digital Elevation Models, vegetation and bodies of water for Digital Earth representations. The proposed system employs a set of sketch-based tools to integrate commonly available data sources, such as orthophotos and Digital Elevation Models (DEM), to facilitate the rapid creation and integration of detailed geospatial content. Consequently, our system can be used to enhance the quality of Digital Earth data by enabling the straightforward creation of new 3D landscape elements.**

I. Introduction

With the recent technological advances in geospatial capturing technologies, there has been increasing interest in the Digital Earth (DE) concept, originally proposed in [1]. DE provides a reference model for the integration, management, visualization and processing of geospatial data. This model efficiently integrates a vast amount of geo-located information such as Digital Elevation Models (DEM), satellite imagery, orthophotos and vector-based features (i.e. road systems). At present, DE software systems already incorporate many of these data-types and are advancing toward supporting 3D contents such vegetation and other landscape elements.

A problem arising in this context is the dynamical nature of our world. This creates a constant demand for the creation and editing of data for DE. Developing interactive tools that support rapid 3D content creation and manipulation for integration into this framework helps to alleviate these demands and complements automatic reconstruction techniques.

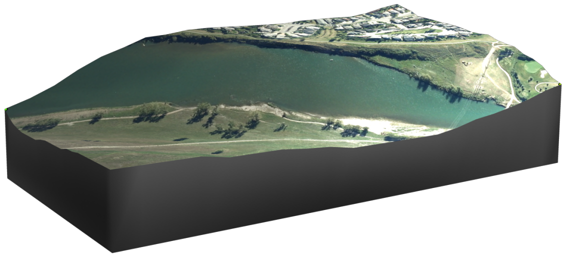


Fig. 1: A distorted river as a result of Imprecise DEM. DEM data is obtained from the *US Geological Survey*.

Geospatial data, within the context of a DE framework, can be broadly categorized into three groups: 2D (i.e. imagery, vector data), 2.5D (i.e. Digital Elevation Model) and 3D (e.g. buildings, bridges, vegetation, bodies of water). Imagery is the most commonly available source of information about the Earth, but does not provide any 3D information. For example, orthophotos, which are aerial photographs with the uniform scale, are available for many regions around the world. In contrast, Digital Elevation Models represents the rough geometry of the Earth's surface and incorporate salient features such as rivers, ridges and hills.

Digital Elevation Models (DEM) are available for the entirety of the Earth's surface. However, the quality and precision of DEM datasets depend on the acquisition techniques employed and varies drastically between datasets [2]. Several factors such as terrain roughness, sampling density, choice of interpolation algorithm, occluded terrain and vertical resolution affect the quality of DEMs [2]. Figure 1 depicts one of the typical issues arising in DEMs generated from low quality data. The characteristic geometry of important terrain features such as rivers, lakes, ridges and cliffs are not necessarily well-represented by DEMs. Therefore, to improve the representation of these features, techniques for improving the accuracy and quality of DEMs is critical.

The 3D models (e.g. vegetation, buildings and bridges) required for detailed DE representations typically do not exist. Additionally, the number and appearance of these objects are continually changing, and nonstop capturing and reconstruction is typically impractical. In recent years, various automatic methods have been proposed for reconstructing terrain and populating them with 3D content [3]. Nevertheless, these methods have a number of limitations. Automatic methods generally have limited robustness which affect the precision of results [3]. For reconstructing a textured 3D object and computing its geographic coordinates, automatic methods typically require geo-referenced high quality input data as well as numerous photos of the object [3]. Moreover, objects have to be clearly visible and non-occluded in photos. In this regard, dense areas like forests and city centres are particularly difficult to reconstruct. Finally, automatic reconstruction methods do not consider scenarios where data is currently unavailable (i.e. landscape planning, historical site reconstruction).

Sketch-based interfaces are a promising paradigm in interactive modeling, offering simple and natural ways to create complex 3D shapes and perform other modeling tasks [4, 5]. However, as observed by Schmidt et al. [6], drawing an accurate shape without assistance can be challenging. Using an image to guide the sketching process helps to create objects quickly and accurately [5, 7, 8]. In addition, the input image and the user sketch provide a model-image correspondence which is particularly useful within the context of our application scenario. Accordingly, in this paper, we introduce a sketch-based modeling system (Figure 2) that uses available

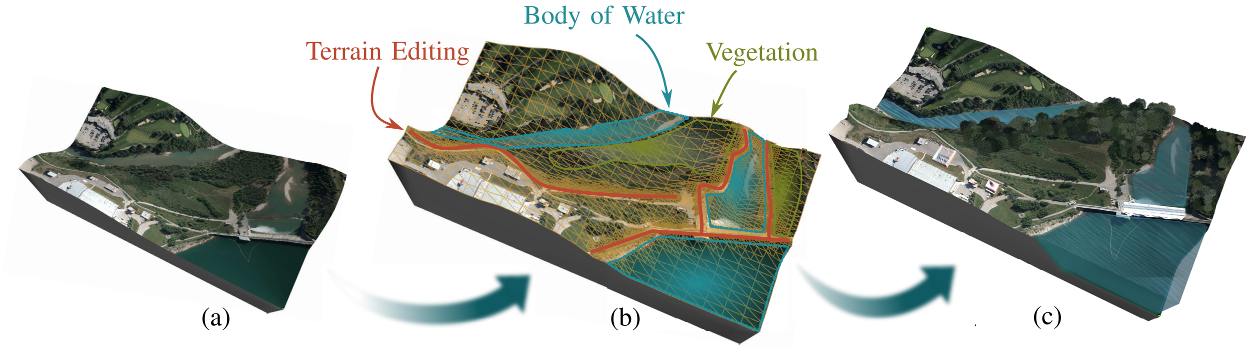


Fig. 2: 3D Maquetter takes as input elevation data and an orthophoto (a). We employ a set of sketch-based tools (b) to create a 3D maquette of the region of interest (c).

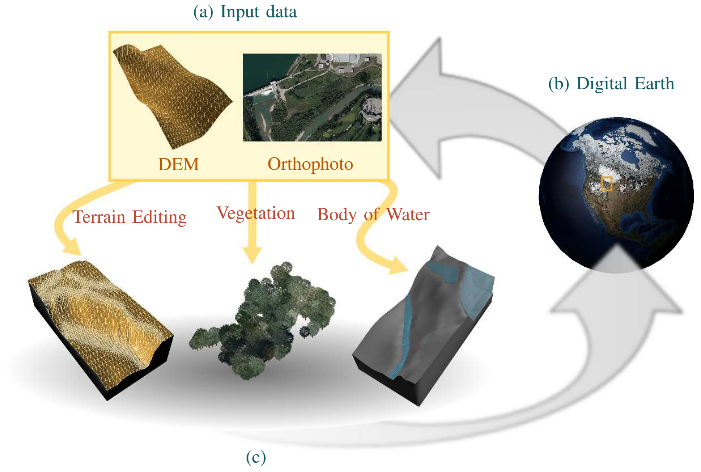


Fig. 3: System Overview: the input data (a) is retrieved from the DE (b). The orthophoto and DEM (a) are used for the creation of the landscape elements (c), and this 3D content is then exported back to the DE.

orthophotos and DEMs to support the rapid creation of textured 3D contents (e.g vegetation, bodies of water) and modification of the terrain geometry. The final result of our system is similar to a 3D *maquette* or miniature model of terrain that includes detailed landscape elements. By taking advantage of orthophotos and DEMs, we provide a suite of image assisted sketch-based modeling tools [7] designed for creating and editing these geospatial models for DE. Our proposed system can thus be used to enhance the quality and availability of current data as well as the creation of new 3D contents.

A. System Overview

Figure 3 illustrates an overview of our system. Our system starts by specifying a region of interest (ROI) in the DE. A ROI is a rectangular area specified by latitudes and longitudes of its corners, or alternatively a cell index of multiresolution reference models of the DE [9, 10]. Our system retrieves the initial input data such as DEM and an orthophoto from the DE framework.

As depicted in Figure 3, various landscape elements may appear in a given orthophoto. To support the creation and editing of 3D content, we thus propose three sketch-based tools supporting the modeling of content based on the most common landscape elements [11] appearing in orthophotos: terrain editor, vegetation and body of water tools. The types of 3D content generated by these tools are illustrated in Figure 3. The terrain editing tool (Section III) facilitates interactive editing of DEM datasets to correct the geometry of the features apparent in an orthophoto, such as rivers, roads and cliffs. The body of water tool (Section IV) interactively generates the volumetric geometry of a body of water. The vegetation tool (Section V) interactively identifies and generates vegetation and plant ecosystems based on orthophotos. As orthophotos are used extensively in our system to texture terrain and guide for modeling, we present the clone tool for modifying and cleaning orthophotos (Section VI). Finally, the integrated result, the 3D maquette, (consists of a textured terrain together with all the created 3D models) are exported back to DE (Figure 3).

B. Contributions

Our main contribution is an image-guided sketch-based system for the rapid creation of 3D content and enhancement of existing content for a DE framework. In the proposed system, we have adapted a number of state-of-the-art techniques, and modified them to address the challenges arising in the creation of 3D models for DE (as discussed in the preceding section). This leads to the following technical contributions: a sketch-based method and corresponding mathematical framework for correcting DEM dataset at multiple resolutions based on *features* visible in orthophotos, as well as an image-based technique for modeling forests and tree stands based on an orthophoto.

II. Related Work

The vision of a Digital Earth as ”a digital replica of the entire planet” was first proposed in Al Gore’s visionary talk on January 1998 [1]. Nowadays, there are several frameworks built based on the concept of Digital Earth. Discrete Global Grid Systems (DGGSs) make such a representation possible by partitioning the Earth’s surface into indexed cells (mostly regular) used to store the data associated with each index [9, 10].

DE frameworks mostly accommodate a variety of 2D and 2.5D geo-spatial data formats and are advancing toward supporting 3D geospatial data. Geographical information systems

(GIS) such as ESRI and BAE systems (SOCET GPX) present various automatic and interactive tools for the creation and editing of geospatial data. These systems support interactive editing of DEM and 2D vector-based features (e.g. roads, bodies of water). However, 2D vector-based features are typically used to visualize various landforms and terrain features (e.g. rivers and roads) on the ground, and they do not have any 3D information. In constrast, the focus of our system is the sketch-based creation and editing of 2.5D and 3D geo-spatial data such as terrain, bodies of water and plants to complement automatic reconstruction techniques. We discuss the modeling of the supported types of geospatial data in the summary of previous work below.

A. *Terrain Editing*

As discussed above, DEM datasets are often low resolution. Consequently, terrain features are not necessarily represented accurately in the underlying DEM. Therefore, interactive image-based tools are essential for the editing of DEM data to accurately represent terrain features, which must be accompanied by the introduction of details at multiple resolutions. Interactive terrain modeling and editing techniques have been the subject of extensive research. Fractal terrain deformation [12] and editing via control handles [13] were common aspects of earlier works. In contrast, direct manipulation methods which offer more natural interaction, are increasingly preferred.

At present, interactive state-of-the-art techniques focus on: brush based, exemplar-based and sketch-based interfaces. Brush based methods [14] present the user with a set of interactive brushes for editing terrain. Although these brushes are well-suited to the sculpting of synthetic terrains, they do not support the editing of pre-existing precise terrain features. Exemplar-based methods [15, 16] edit terrain by finding the most similar region to a given area. However, terrain features have unique characteristics and geometry which makes matching non-trivial and error-prone.

The tool we propose is more closely related to sketch-based approaches. Sketch-based methods have been widely used for editing terrain, and can be divided into two categories based on the viewpoint used to provide input. First person sketch-based systems [17, 18] introduce interactive methods for editing terrain from a profile view. These methods provide limited control over the deformation of features.

Alternatively, interfaces also permit users to edit terrain from a number of different viewpoints. One such approach was presented by Gain et al. [19] who proposed a sketch-based technique for modeling synthetic terrain at a single resolution. However, precise editing of terrains based on features, such as rivers and cliffs with various slopes, was somewhat tedious and required multiple interactions. Bernhardt et al. [20] suggest a sketch-based method for deforming terrain based on features defined by elevation constraints. Their choice of constraint forced all features to have the same slope, in disagreement with real terrains. In contrast, our approach is a unique image assisted sketch-based method which allows the terrain to be modified based on features obtained from orthophotos. Terrains can be edited freely from any point of view and the slopes are adjusted using a single stroke specifying the cross section of the terrain.

B. *Vegetation*

Plants are a ubiquitous part of urban areas and landscapes. Adding vegetation to DE frameworks increases their accuracy, as well as the realism of their visualization. Our system generates plant ecosystem based on an orthophoto using a sketch-based tool for specifying the areas covered with larger vegetation, such as shrubs and trees. In the literature, a number of methods have been proposed for generating trees and plant ecosystem which are either image-assisted or procedural.

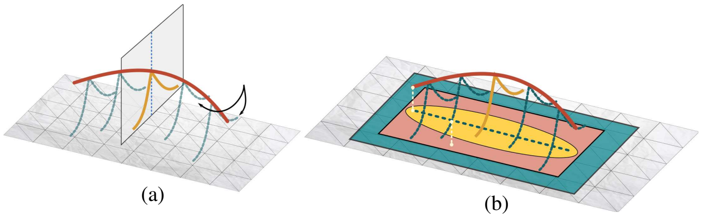
Existing literature on generating trees using procedural modeling is vast. Photographs have also been used for modeling trees [21, 22]. These methods are designed to model a tree from either a single image or multiple images. Our work is, however, most related to generating plants ecosystem. Simulation-based methods [23, 24] have been extensively employed to generate forests and urban ecosystems. Hammes [25] proposed a technique for generating ecosystem based on DEMs. At the same time, the result of these methods may not be consistent with an orthophotos of a modeled region.

Although orthophotos are available for most regions of the earth's surface, their quality and viewpoint make them ineffective for reconstructing plant ecosystems automatically. Some methods have been proposed for counting trees in an orthophoto [26, 27]. In contrast, we propose a sketch-based data-driven method for generating vegetation based on an orthophoto. Our method combines both procedural and imaged-based techniques to generate a plant ecosystem consistent with a given orthophoto. Distributing plants onto the terrain and coloring them based on an orthophoto are done similar to previously proposed procedural modeling and imaged-based techniques, respectively [24, 21].

III. Terrain Editing Tool

Orthophotos provide information about a variety of natural and man-made *features* such as rivers, cliffs, ridges and roads. Each of these features has unique characteristics that affect the geometry of the terrain (i.e. elevation, slopes, orientation). However, current DEM datasets are typically not detailed enough to accurately capture these features. Here, we introduce a sketch-based terrain editing tool to address this problem by identifying visually apparent features of the orthophotos. The geometry of features is defined by a *control curve*, elevation along the curve, slopes and fields of influence guided by the orthophoto (Figure 4a). The orange curves in Figure 4a illustrate an example of the left and right slopes around a feature specified by a *cross section curve*. The length of these strokes specifies the feature's field of influence. As depicted in Figure 5, a variety of features can be represented by simply changing the form of these two curves.

Features generated using this method are typically more detailed than the highest resolution of existing DEM datasets. Therefore, to correct the geometry of features precisely in the terrain, the resolution around these features must be increased. Accordingly, we introduce a multiresolution terrain editing method. This method iteratively modifies the terrain from low to high resolutions to fit a set of *positional* and *energy minimization* constraints. These constraints are created from the input strokes provided by the user. To increase computational efficiency, we adaptively subdivide the base

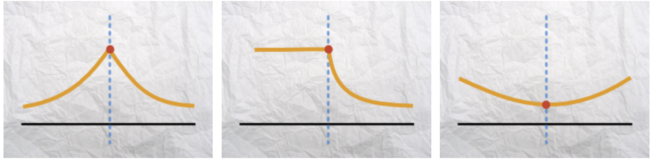


(a)

(b)

Fig. 4: Specification of a feature from a control curve. (a) The geometry of a feature is specified by the control (red curve) and cross section curve (orange curve). The cross section curve is placed at regular intervals along the control curve (blue curves). (b) The vertices within the yellow and pink regions are displaced to satisfy the positional constraints imposed based on the control and cross section curves, respectively. The energy minimization constraints are imposed on all the vertices within the blue region.

terrain near features. The details of our method are provided in the remainder of this section, where Section III-A introduces our interface for sketch-based interaction, and Section III-B describes a deformation technique based on the input features this interface generates.



(a) Ridge

(b) Cliff

(c) River bed

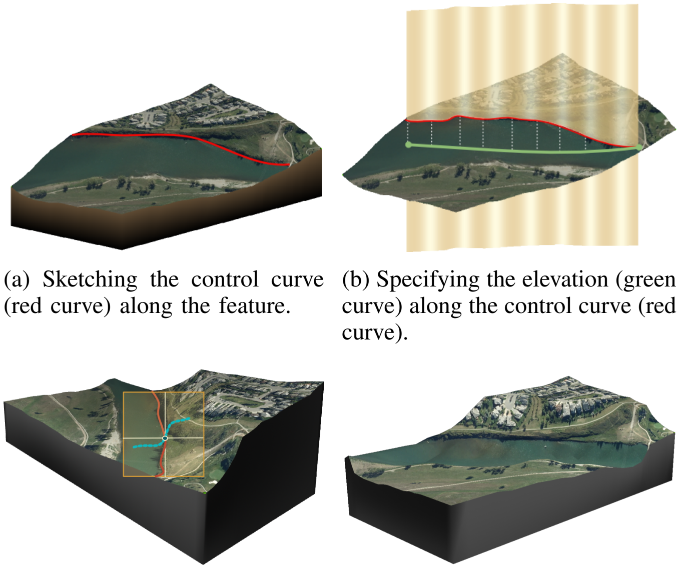
Fig. 5: Various terrain features can be represented by changing the slopes around the feature.

A. Sketch-based Interaction

A feature's geometry is determined by three strokes which specify: a *control curve*, the elevation along the control curve and a *cross section curve* (Figure 4a). First, the user sketches a control curve onto the terrain (Figure 6a). The initial elevation of the control curve is then determined by the control curve's projection onto the terrain. To change the control curve's elevation, a *curtain* is automatically generated for sketching the elevation profile along the curve (Figure 6b). To control the feature's slopes and fields of influence, the cross section curve (Figure 4a) is sketched on two sides of the control curve (Figure 6c). Finally, the terrain is deformed to best match the control and cross section curve (Figure 6d).

B. Feature-based Multiresolution Terrain Deformation

Digital Elevation Models are stored in two formats: *height map* and *triangular irregular network (TIN)* [2]. Due to its simplicity and computational efficiency, height maps have become the most prevalent format for representing DEM. In addition, preserving the regularity of the multiresolution terrain in the height map format is more challenging than TIN. Thus, our tool retrieves DEM from DE in the height map format. To capture the details of input features, we employ subdivision methods for increasing the resolution of the underlying DEM. To support both DEM formats, we use Loop subdivision [28], as suggested by [29], for this task by dividing each rectangular cell into two triangles. To export the modified DEM back into DE, several resolutions of DEM data must be stored in the height map format. To address this issue, we propose a hierarchical representation of the terrain resulted from the subdivision method. Therefore, given a terrain at the base resolution, we correct the geometry of the terrain at several resolutions, and we store each resolution in a height map which best fits the input features.



(a) Sketching the control curve (red curve) along the feature.

(b) Specifying the elevation (green curve) along the control curve (red curve).

(c) Specifying the slopes and fields of influence by sketching the cross section curve (blue curve).

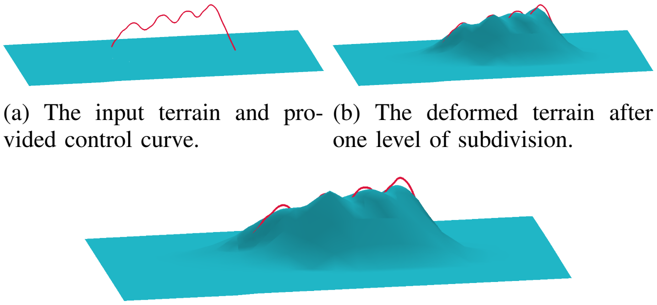
(d) The geometry of the river's edge is corrected based on the feature.

Fig. 6: Deformation tool.

As described in Section III-A, strokes are divided into two groups: control and cross section curves. Our goal is to deform the input terrain to best match control and cross section curves while preserving other characteristics of terrain. As the terrain is stored in a hierarchical representation, our algorithm has to support terrain deformation at different resolutions. To explain the terrain deformation technique based on input features, first we describe a method for approximating a single control curve, and then we present the terrain deformation technique that operates on multiple features.

1) Terrain deformation based on a single control curve: Given a terrain T with the base resolution T_0, we develop a multiresolution terrain deformation technique such that the terrain at resolutions $\{0, 1, \ldots, k\}$ best fit the given control curve. The control curve is defined by the polyline constructed from a set of 3D points denoted as $P = \{p_1, p_2, \ldots, p_m\}$ captured from the input stroke.

Various methods have been proposed for surface deformation [30]. Pusch and Samavati [31] introduce a technique which supports both locality and multi-resolution nature of our problem. They present a general framework for local constraint-based subdivision surface deformation. By starting from a given subdivision surface and a set of positional constraints, they solve a weighted least-squares problem to determine the control polygon of the subdivision surface. Although their method supports terrain deformation at multiple resolutions, it is unable to approximate a control curve more detailed than the initial terrain. Figure 7 illustrates an example of the terrain deformation based on the given control curve. Since the curve is more detailed than the initial terrain, displacing the original vertices is not enough to accurately approximate the control curve at higher resolutions. Therefore, we extend their method to accurately approximate detailed control curves.


(a) The input terrain and provided control curve.

(b) The deformed terrain after one level of subdivision.

(c) The deformed terrain after three levels of subdivision.

Fig. 7: The terrain deformation technique, proposed by [31], applied to multiple resolutions. As the input control curve is more detailed than the initial terrain, displacing the original vertices is not enough to accurately approximate the control curve at higher resolutions.

To provide a good fit for detailed curves, our technique must capture the curve's details at several resolutions. To address this issue, we not only move the initial vertices, but also solve an optimization problem for the vertices replacements at each resolution to capture the curve's fine details. Therefore, as the terrain's resolution increases, the terrain approximates details which could not be captured at the lower resolutions. Accordingly, for each resolution t, given the terrain T_t, we place the vertices V^t such that it minimizes the distance between the control curve and the subdivided terrain:

$$\min_{\Delta_t} \; d(S(T_t + \Delta_t), P) \tag{1}$$

where Δ_t is a perturbation vector for V^t, $S(T)$ denotes subdivision of T, and d is the distance between P and the subdivided terrain. The distance between the subdivided terrain and the points $p_j \in P$ of the control curve is computed using the distance between p_j and its projection p_j^{t+1} onto the subdivided terrain. The projection p_j^{t+1} falls inside a triangle with vertices v_a^{t+1}, v_b^{t+1} and v_c^{t+1} and can be written as:

$$p_j^{t+1} = \alpha v_a^{t+1} + \beta v_b^{t+1} + \gamma v_c^{t+1} \tag{2}$$

where α, β and γ are the barycentric coordinates. Therefore, to minimize Equation 1, we minimize $\sum \| p_j^{t+1} - p_j \|$ for $p_j \in P$ where p^{t+1} is defined in Eq. 2. This produces the following positional constraints:

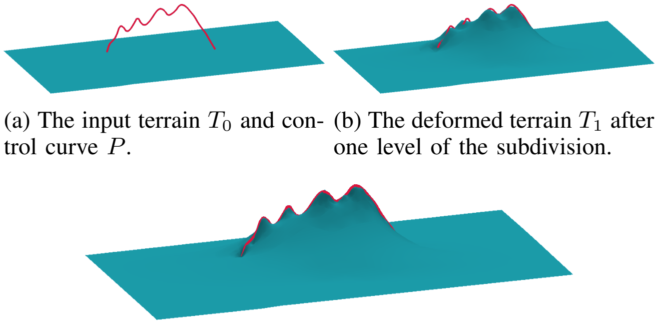
$$p_j^{t+1} = p_j, \text{ for } j \in \{1, 2, \ldots, m\}. \tag{3}$$

which, we can rewrite as a function of V^t using Eq. 2. As our subdivision mask is a linear operator, the position of every vertex v_i^{t+1} can be written as $v_i^{t+1} = \alpha_1 v_1^t + \alpha_2 v_2^t + ... + \alpha_n v_n^t$, where n is the number of vertices at resolution t, and the coefficient α_j is defined by S_i (i.e. the ith row of the subdivision matrix S). Therefore, a positional constraint can be rewritten to depend on the vertices V^t:

$$\begin{aligned} p_j^{t+1} &= \alpha v_a^{t+1} + \beta v_b^{t+1} + \gamma v_c^{t+1} \\ &= \alpha S_a V^t + \beta S_b V^t + \gamma S_c V^t \\ &= [\alpha S_a + \beta S_b + \gamma S_c] \begin{bmatrix} v_1^t & v_2^t & v_3^t & \ldots & v_n^t \end{bmatrix}^T , \end{aligned} \tag{4}$$

yielding a banded linear system of equations relating P and V^t.

The positional constraints form an overdetermined system, and the minimizer of this system is computed by solving a least-squares problem (i.e using the pseudo-inverse). Figure 8 shows an example of employing this method to deform a flat terrain. In the next sections, the preceding method is extended to a set of specified features.


(a) The input terrain T_0 and control curve P.

(b) The deformed terrain T_1 after one level of the subdivision.

(c) The deformed terrain T_3 after three levels of the subdivision.

Fig. 8: The terrain deformation technique at multiple resolutions.

2) Terrain deformation based on a set of features: For this case, the geometry of features is specified by control and cross sectional curves. Our goal is to deform the terrain to best fit the control curves, associated slopes and fields of influence. Figure 4a depicts an input feature in which the control curve (red curve) specifies the feature and elevation along it, and the cross section curve (orange curve) specifies the feature's slope and field of influence. Here we extend the previous method to approximate not only the control curve, but also a feature slopes and field of influence. To approximate the slope and field of influence along the control curve, we define extra positional constraints based on the cross section curve. To impose these constraints along the control curve, the cross section curve is translated and scaled at regular intervals along the control curve, and oriented using the rotation minimizing frame [32] as shown in Figure 4a. Thus, given a generated cross section curve defined by the polyline constructed from a set of 3D points denoted as $C = \{c_1, c_2, \ldots, c_l\}$, extra positional constraints can be defined as:

$$c_j^{t+1} = c_j, \text{ for } j \in \{1, 2, \ldots, l\}, \tag{5}$$

where c_j^{t+1} is the projection of c_j onto the subdivided terrain (Eq. 2) (Figure 4b).

The control and cross section curves do not have the same importance in the resulting least-squares problem, as the control curve is more accurately specified in orthophotos.

To address this issue, we impose the positional constraints of the cross section curves after determining locations of vertices based on the control curve, as suggested by Hnaidi et al. [33]. This gives rise to two least-squares problems which determine the positions of V^t. In the first problem, we compute the positions of vertices V^t that are affected by the constraints defined in Eq. 3, and in the second problem, by fixing the positions of the vertices in the previous step, we compute the positions of vertices based on Eq. 5 (Figure 4b). Additionally, dividing the problem into two subproblems reduces the size of the least-squares problem and increases computational efficiency.

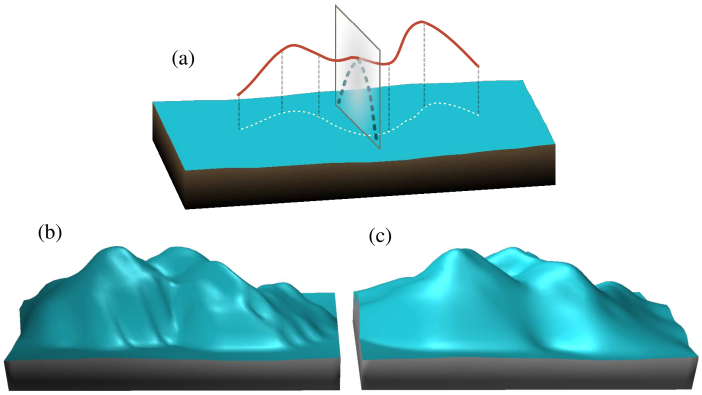


Fig. 9: An example of the terrain deformation with and without energy minimization constraints. (a) The input feature and terrain. (b) The deformed terrain only based on the positional constraints. (c) The deformed terrain using the positional and energy minimization constraints.

The editing of the terrain purely based on positional constraints can result in high curvature areas due to non-regularized least-squares solutions [31]. Furthermore, moving a subset of the terrain's vertices without considering the adjacent vertices can result in high curvature areas around the boundary of the deformed region (Figure 9). To address these issues, we introduce a constraint to minimize the curvature of the deformed region. To approximate surface curvature at a vertex, we use the discrete Laplace-Beltrami operator [34]. Thus, a energy minimization constraint for each vertex v_j^{t+1} is defined as:

$$L_j^{t+1} = v_j^{t+1} - \frac{1}{d_j} \sum_{v_i^{t+1} \in N(v_j^{t+1})} v_i^{t+1} = 0,$$

where d_j and $N(v_j^{t+1})$ are the degree and adjacent vertices of v_j^{t+1}. To eliminate high energy behaviors, we impose the energy minimization constraint on all the vertices that are affected by the control and cross section curves or falls within a specified distance from the control curve (Figure 4b). The above constraint is considered along with the positional constraints for the vertices relocated by both least-squares problem.

As mentioned earlier, to approximate features and associated characteristics at each resolution, our technique must be iteratively applied to the result of optimizing the previous resolution to reach adequate precision. By applying the method repeatedly, as the number of vertices at the base terrain and the size of least-square system increase, we obtain higher accuracy around features. Finally, this approach can be extended to a set of features by computing the positional and energy minimization constraints based on all control and cross section curves.

Since features may only affect a small region on DEM, we avoid increasing the resolution for the entire region of interest. To increase details around features, we adpatively subdivide the terrain (Figure 10) by employing incremental adaptive Loop subdivision [35]. As discussed by Pakdel and Samavati, adaptive subdivision techniques have some shortcomings which must be handled delicately [35]; otherwise, skinny triangles, cracks or abrupt change of resolution may appear in the resulting DEM. Continuous change of details makes rendering DEMs at different resolution possible with a simple and efficient technique such as zero area triangles [36]. In our system, to preserve the hierarchy of DEM and support continuous change of details, terrain is adaptively subdivided such that adjacent triangles must be within one level of each other in the terrain hierarchy.

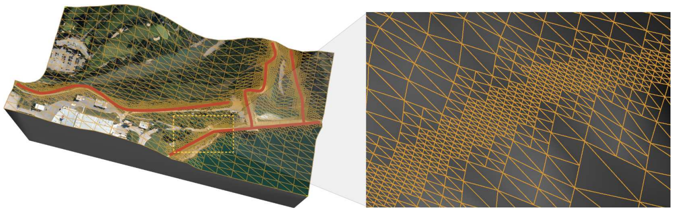


Fig. 10: An example showing the adaptively subdivided terrain based on the features. The left image illustrates the provided features on the terrain, and the right one depicts the adaptively subdivided terrain around a feature.

IV. Bodies Of Water Tool

Representing bodies of water is important for many DE environmental applications which need monitoring, visualizing and simulating water bodies. However, acquisition techniques of Digital Elevation Model are mostly unable to capture the underlying structure of rivers, lakes and sea beds. In our system, bodies of water can be interactively created using a simple sketch-based tool. To use this tool, first terrain has to be edited to create a basin (see Figure 11). Second, the user draws a closed stroke onto the terrain corresponding to the water body boundary. Our system then automatically generates the body of water based on the elevations of vertices inside and around the region enclosed by the stroke.

V. Vegetation Tool

Orthophotos provide some information regarding the plant ecosystems present in a given terrain, and augmenting DE representations with plant models substantially increases their accuracy and realism in 3D scenes. However, these photos are commonly insufficient for the 3D reconstruction of individual trees and their ecosystem. On the other hand, they provide vast amounts of information about the placement, distribution and color of plants. Accordingly, our system provides a sketch-based tool guided by an orthophoto for creating vegetation on terrain. As initial data, 3D models of trees and plants are retrieved from a database (in a DE this would be based on the region of interest or commonly available vegetation species

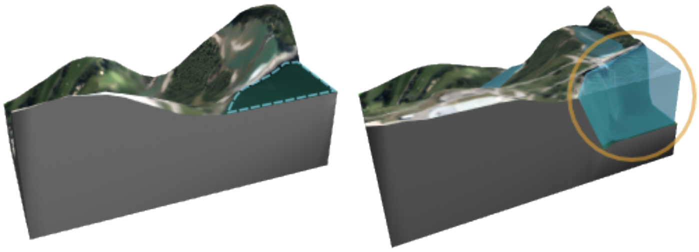


Fig. 11: An example showing the application of the body of water tool. The left image shows the input terrain and the boundary of the body of water, and the right image depicts the body of water created by employing our tools.

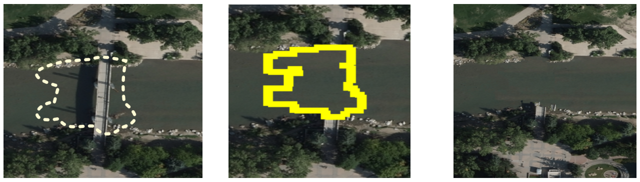
diversity). As shown in Figure 12, the region containing plants and vegetation is specified interactively by sketching a closed stroke onto the terrain based on the orthophoto (cyan stroke). Similar to [24], plants are distributed onto the region based on the average distance between the input plants in the region. The average distance can be provided either automatically [27] or interactively by the user. The created plants are colored based on the orthophoto to create a plant ecosystem with a similar visual character to that present in the selected region. Therefore, the top view of the terrain with vegetation remains consistent with the orthophoto (see Figure 12).

To distribute plants onto the specified area, we start by projecting strokes onto the terrain, and triangulating the 3D polygon with respect to DEM data using *Delaunay trianglua-tion* [37]. Afterwards, plants are randomly distributed onto the region with respect to the areas of triangles (i.e. larger triangles get more plants than smaller ones). The number of plants for each input model is determined based on the average distance specified for the region. Finally, leaves are colored based on the orthophoto. Our tool considers small neighborhood around each position to determine the leaves' color. Furthermore, plants of the same type are randomly scaled and rotated to create more variation.

VI. Terrain Texturing using Orthophotos

Orthophotos contain a vast amount of information regarding the visual appearance of features. Consequently, using them to texture the terrain enhances its visual appearance greatly. Since many features are represented in orthophotos, it is particularity beneficial to have a set of smart image editing tools to modify them. Our system includes a clone brush for editing these images. The clone brush can be used for removing unwanted regions. For instance, a 3D object such as a bridge is not part of the geometry of the terrain, so its footprint and shadow must be removed and replaced by terrain material to be used as texture (see Figure 13). Clone brush can also be used for repairing the texture of objects when they are obscured by occlusion.

This tool can also be used for cloning features, such as vegetation or bodies of water, to create a new image which can be used later as a guide for modeling new landscapes (see Figure 14). Figure 14a and 14b illustrate the original and modified image, respectively. As demonstrated in Figure 14b, using our clone brush, landscape elements have been modified to create a new scene. Finally, Figure 14c presents the result after creating new landscape elements interactively using our tools. This feature is particularly beneficial for landscape planning applications.

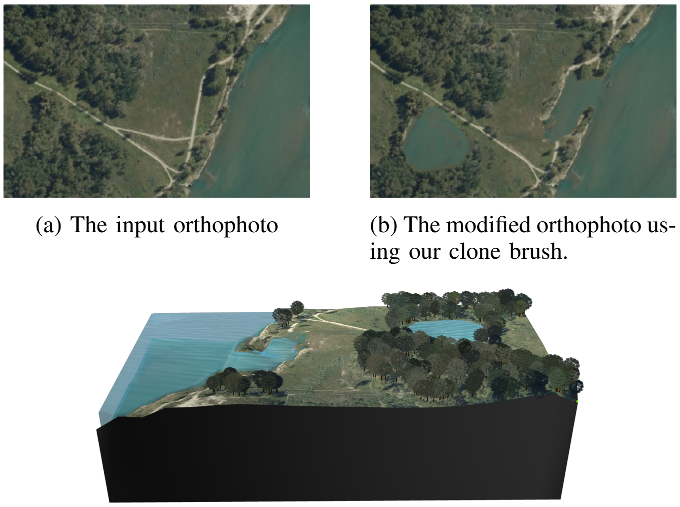


(a) Sketching the unwanted region in the original image.

(b) The modified image after cloning. All the pixels up to a specific distance d from the boundary are colored in yellow.

(c) The final result after synthesizing the boundary pixels of the unwanted region.

Fig. 13: An example showing the application of Clone brushing tool for removing unwanted regions.



(a) The input orthophoto

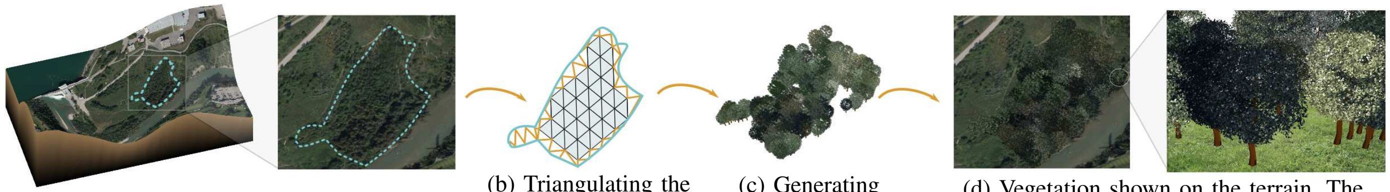
(b) The modified orthophoto using our clone brush.

(c) The result after generating the new landscape from the modified photo.

Fig. 14: A novel landscape generated on the basis of a preexisting terrain.

There are two challenges regarding cloning a portion of an image to another region. Copying information from one part of an image to another can result in distortion at the boundary of the selected region. To minimize distortion around the boundary, all the pixels up to a specific distance d from the boundary are synthesized based on the inside and outside regions (see Figure 13b). To have a fast real-time tool, we use the texture synthesizer proposed by Simakov [38], and for finding the best patch we apply PatchMatch [39].

Furthermore, our clone brush considers the underlying terrain geometry as opposed to the traditional image processing tool. For instance, sloped terrain causes texture foreshortening. Therefore, to avoid unrealistic distortion during cloning, our tool adaptively resizes the destination region based on terrain slopes at the source and destination.



(a) The input stroke around the region covered with vegetation.

(b) Triangulating the projected stroke with respect to DEM data.

(c) Generating plants based on the orthophoto and selected region.

(d) Vegetation shown on the terrain. The top view of the terrain with vegetation remains consistent with the orthophoto. The right image shows a closer view of the created scene.

Fig. 12: An example demonstrating the vegetation tool.

VII. Results

To illustrate the methods presented in previous sections, we implemented a sketch-based system which supports a variety of landscapes. As input data, we used DEMs available from the *US Geological Survey*, and orthophotos from the *City of Calgary* datasets. We present an example of the creation and editing of landscape by considering the Glenmore reservoir located in the southwest quadrant of Calgary, Alberta (Figure 15). The input data and generated contents are individually depicted in Figure 3.

Figure 15a and 15b illustrate the input orthophoto and terrain, respectively. Features such as the river, reservoir and roads in the orthophoto are not accurately represented in the input DEM. To correct the input DEM, four features are specified in the orthophoto. To create bodies of water, three closed stroke are sketched onto the terrain. Vegetation is created based on four strokes around the areas covered by plants. The final result is illustrated in Figure 15c and 15d. To export this information for a height-map based DE framework, the terrain hierarchy is represented in height map format for each resolution.

Figure 16 depicts another example of the creation and editing of landscape by considering Elliston lake located in the southeast quadrant of Calgary, Alberta (Figure 16). Figure 16a and 16b illustrate the input orthophoto and terrain, respectively. Some feature including the lake and roads are not accurately represented in the underlying DEM data. Therefore, to correct the geometry of the terrain, two features (the lake and one of the roads) are specified in the orthophoto, and the body of water is created by sketching a closed stroke around the boundary of the lake. Vegetation is specified and created based on six strokes around the areas covered by plants.

Additionally, our system supports the integration of new designs and ideas into a DE representation. As illustrated in Figure 14, by modifying an orthophoto using our tools, a new landscape can be modeled and explored in 3D. Our system creates a platform for the setup, analysis and visualization of new concepts within the context of DE. Our supplementary materials including a video provide more information about our system.

VIII. Conclusion

In this paper, we introduce a sketch-based system for creating 3D contents from a single photo and enhancing the quality of existing data in a DE framework. Our system is capable of creating a wide range of landscape from limited input data, such as low quality DEM and an orthophoto.

There are several directions which this work can be extended. To evaluate our system by users, a formal user study could be conducted. For generating plants ecosystem based on an orthophoto, the density of plants could potentially be obtained via frequency analysis of an orthophoto. Currently, the user provides the average distance between plants in the photo. To make our system simpler and more interactive, it can support snapping and flood fill operations for specifying features such as rivers and edges of structures [7]. Additionally, our system can be extended to support the sketch-based creation of other objects (e.g. buildings and bridges) from orthophotos.

References

[1] A. Gore, “The digital earth: understanding our planet in the 21st century,” *Australian surveyor*, vol. 43, no. 2, pp. 89–91, 1998.

[2] Z. Li, Q. Zhu, and C. Gold, *Digital terrain modeling: principles and methodology*. CRC press, 2010.

[3] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. Gool, and W. Purgathofer, “A survey of urban reconstruction,” in *Computer Graphics Forum*, vol. 32, no. 6. Wiley Online Library, 2013, pp. 146–177.

[4] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge, “Sketch-based modeling: A survey,” *Computers & Graphics*, vol. 33, no. 1, pp. 85–103, 2009.

[5] L. Olsen, F. Samavati, and J. Jorge, “Naturasketch: Modeling from images and natural sketches,” *Computer Graphics and Applications, IEEE*, vol. 31, no. 6, pp. 24–34, Nov-Dec 2011.

[6] R. Schmidt, A. Khan, G. Kurtenbach, and K. Singh, “On expert performance in 3d curve-drawing tasks,” in *Proceedings of the 6th eurographics symposium on sketch-based interfaces and modeling*. ACM, 2009, pp. 133–140.

[7] L. Olsen and F. F. Samavati, “Image-assisted modeling from sketches,” in *Proceedings of Graphics Interface 2010*, ser. GI ’10, 2010, pp. 225–232.

[8] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or, “3-sweep: extracting editable objects from a single photo,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 195, 2013.

[9] M. F. Goodchild, “Discrete global grids for digital earth,” in *Proceedings of 1 st International Conference on Discrete Global Grids, March*, 2000.

[10] A. Mahdavi-Amiri, E. Harrison, and F. Samavati, “Hexagonal connectivity maps for digital earth,” *International Journal of Digital Earth*, vol. 0, no. 0, pp. 1–20, 2014.

[11] R. M. Smelik, T. Tutenel, R. Bidarra, and B. Benes, “A survey on procedural modelling for virtual worlds,” vol. 33, no. 6, 2014, pp. 31–50.

[12] S. Stachniak and W. Stuerzlinger, “An algorithm for automated fractal terrain deformation,” *Computer Graphics and Artificial Intelligence*, vol. 1, pp. 64–76, 2005.

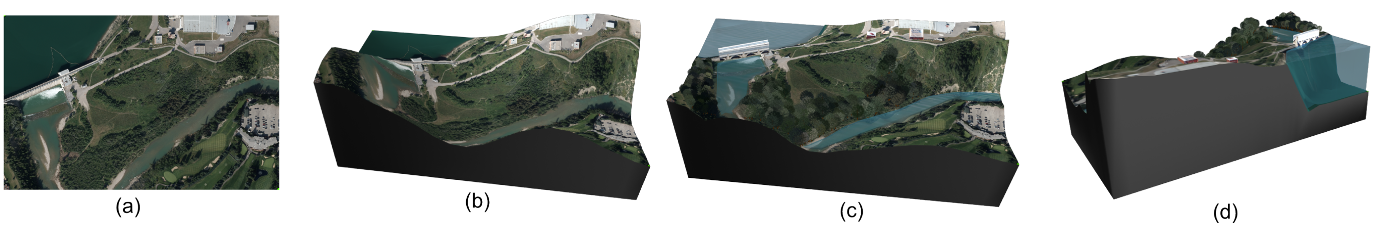


Fig. 15: Glenmore reservoir. (a) Input orthophoto. (b) Original terrain. (c) Result after using the proposed system. (d) Result from a different viewpoint.

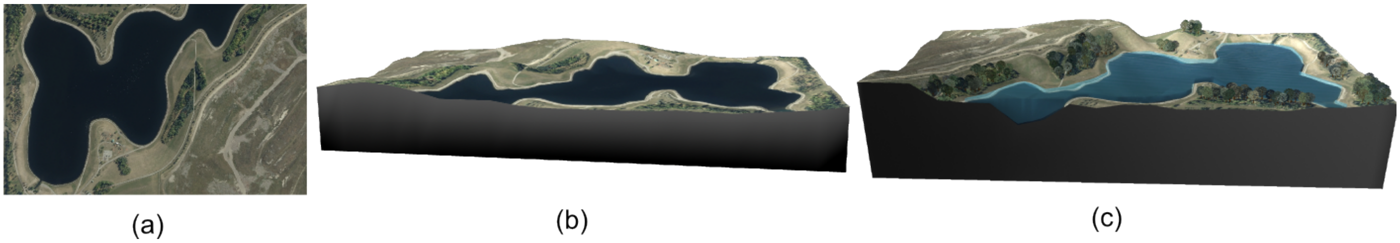


Fig. 16: Elliston Regional Park. (a) Input orthophoto. (b) Original terrain. (c) Result after using the proposed system.

[13] H. Jenny, B. Jenny, W. E. Cartwright, and L. Hurni, “Interactive local terrain deformation inspired by hand-painted panoramas,” *The Cartographic Journal*, vol. 48, no. 1, pp. 11–20, 2011.

[14] G. J. de Carpentier and R. Bidarra, “Interactive gpu-based procedural heightfield brushes,” in *Proceedings of the 4th International Conference on Foundations of Digital Games*. ACM, 2009, pp. 55–62.

[15] H. Zhou, J. Sun, G. Turk, and J. M. Rehg, “Terrain synthesis from digital elevation models,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 4, pp. 834–848, 2007.

[16] J. Brosz, F. Samavati, and M. Sousa, “Terrain synthesis by-example,” in *Advances in Computer Graphics and Computer Vision*, 2007, vol. 4, pp. 58–77.

[17] F. P. Tasse, A. Emilien, M.-P. Cani, S. Hahmann, and N. Dodgson, “Feature-based terrain editing from complex sketches,” *Computers & Graphics*, vol. 45, pp. 101–115, 2014.

[18] V. A. dos Passos and T. Igarashi, “Landsketch: A first person point-of-view example-based terrain modeling approach,” in *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*. ACM, 2013, pp. 61–68.

[19] J. Gain, P. Marais, and W. Straßer, “Terrain sketching,” in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*. ACM, 2009, pp. 31–38.

[20] A. Bernhardt, A. Maximo, L. Velho, H. Hnaidi, and M.-P. Cani, “Real-time terrain modeling using cpu-gpu coupled computation,” in *24th SIBGRAPI Conference*. IEEE, 2011, pp. 64–71.

[21] P. Tan, T. Fang, J. Xiao, P. Zhao, and L. Quan, “Single image tree modeling,” in *ACM Transactions on Graphics (TOG)*, vol. 27, no. 5. ACM, 2008, p. 108.

[22] S. Longay, A. Runions, F. Boudon, and P. Prusinkiewicz, “Treesketch: interactive procedural modeling of trees on a tablet,” in *Proceedings of the international symposium on sketch-based interfaces and modeling*. Eurographics Association, 2012, pp. 107–120.

[23] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz, “Realistic modeling and rendering of plant ecosystems,” ser. SIGGRAPH ’98. ACM, 1998, pp. 275–286.

[24] B. Lane, P. Prusinkiewicz *et al.*, “Generating spatial distributions for multilevel models of plant communities,” in *Graphics Interface*. Citeseer, 2002, pp. 69–80.

[25] J. Hammes, “Modeling of ecosystems as a data source for real-time terrain rendering,” in *Digital Earth Moving*. Springer, 2001, pp. 98–111.

[26] F. Santoro, E. Tarantino, B. Figorito, S. Gualano, and A. M. D’Onghia, “A tree counting algorithm for precision agriculture tasks,” *International Journal of Digital Earth*, vol. 6, no. 1, pp. 94–102, 2013.

[27] L. Yang, X. Wu, E. Praun, and X. Ma, “Tree detection from aerial imagery,” in *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2009, pp. 131–137.

[28] C. Loop, “Smooth subdivision surfaces based on triangles,” 1987.

[29] D. Zorin, P. Schröder, and W. Sweldens, “Interactive multiresolution mesh editing,” in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 259–268.

[30] M. Botsch and O. Sorkine, “On linear variational surface deformation methods,” *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 1, pp. 213–230, 2008.

[31] R. Pusch and F. Samavati, “Local constraint-based general surface deformation,” in *Shape Modeling International Conference (SMI) 2010*, 2010, pp. 256–260.

[32] W. Wang, B. Jüttler, D. Zheng, and Y. Liu, “Computation of rotation minimizing frames,” *ACM Transactions on Graphics (TOG)*, vol. 27, no. 1, p. 2, 2008.

[33] H. Hnaidi, E. Guérin, S. Akkouche, A. Peytavie, and E. Galin, “Feature based terrain generation using diffusion equation,” in *Computer Graphics Forum*, vol. 29, no. 7. Wiley Online Library, 2010, pp. 2179–2186.

[34] M. P. Do Carmo and M. P. Do Carmo, *Differential geometry of curves and surfaces*. Prentice-hall Englewood Cliffs, 1976, vol. 2.

[35] H.-R. Pakdel and F. Samavati, “Incremental adaptive loop subdivision,” in *Computational Science and Its Applications–ICCSA 2004*. Springer, 2004, pp. 237–246.

[36] F. Losasso and H. Hoppe, “Geometry clipmaps: terrain rendering using nested regular grids,” in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 769–776.

[37] M. De Berg, M. Van Kreveld, M. Overmars, and O. C. Schwarzkopf, *Computational geometry*. Springer, 2000.

[38] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, “Summarizing visual data using bidirectional similarity,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008.* IEEE, 2008, pp. 1–8.

[39] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, “Patchmatch: A randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.